

Facebook Hacker Cup 2018 Qualification Round

[Request Clarification](#)

Ethan Searches for a String

45 points

Last valid submission time: 2018年7月9日 at 3:38 [\[details\]](#)

[Download Input](#)

[Choose Output](#) No output file selected

[Submit](#)

Ethan's doing his very first programming assignment: implementing a `contains()` function. This function takes two strings, **A** and **B**, and returns `true` if **A** is a substring of **B**, and `false` otherwise.

Here's the algorithm that Ethan has come up with. Note that $|A|$ denotes the length of **A**, and the individual characters of the strings are 1-indexed.

1. Set i and j to each be equal to 1.
2. If $i > |A|$, return `true`.
3. If $j > |B|$, return `false`.
4. If $A_i = B_j$, increment i and j by 1 each, and return to Step 2.
5. If $i = 1$, increment j by 1, and return to Step 2.
6. Set i to be equal to 1, and return to Step 2.

As the TA in charge of grading Ethan's assignment, this doesn't look quite right to you. To make sure Ethan doesn't get any more credit than he deserves, you'd like to find some inputs for which his algorithm returns `false` even though it should return `true`.

The professor teaching this class has provided you with a half-written list of test cases. In particular, it's a list of inputs for the **A** parameter, and you're free to come up with your own inputs for the **B** parameter. For each given string **A**, you want to find a string **B** that will cause Ethan's algorithm to return the wrong output (`false` instead of `true`), if possible. **A** will only contain uppercase alphabetic characters, and **B** must follow the same constraint. The test cases shouldn't be too large, so **B** must also contain at most 10,000 characters.

Input

Input begins with an integer **T**, the number of given strings. Then, **T** lines follow. Each line contains a single string, **A**.

Output

For the i th given string, print a line containing "Case i :" followed by any valid string **B** that will cause Ethan's algorithm to return the wrong value, or "Impossible" if no such string exists.

Constraints

$1 \leq T \leq 100$
 $1 \leq |A| \leq 2,000$

Explanation of Sample

In the first case, i and j will have these values in order the first 10 times the algorithm is at Step 2:

```
i j
---
1 1
2 2
1 2
1 3
1 4
1 5
2 6
3 7
4 8
1 8
```

Please note that other outputs for example cases 1 and 3 would also be accepted.

Sample input · [Download](#)

Sample output · [Download](#)

```
4
ABACUS
FACEBOOK
XYZXYZYX
FBFBF
```

```
Case #1: ASUCABABACUSA
Case #2: Impossible
Case #3: XZYXZYXZYXZYXZYXZYX
Case #4: Impossible
```



This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](#).



Chan Lam Lao

溫郁婷

郭曜禎

● 聊天室(14)

